SYNTAX ZX80®

A PUBLICATION OF **THE HARVARD GROUP**

VOL. 2 NO. 9 ISSN 0273-2696 SEPT., 1981

8K ROM UPDATE

Sinclair now has another source of 8K ROMs. According to Nigel Searle, the new manufacturer will deliver by September 1 enough ROMs to both replace bad ROMs and satisfy demand for new ones.

Searle added that Sinclair opted for a new supplier because European demand for ZX81s was using up all available 8K ROMs.

SINCLAIR FALL COMPUTER SHOW

Sinclair Research plans to hold computer shows to introduce its ZX81 this fall. Response to the first show, to be held in Boston, will determine later sites. No dates have been announced, but Sinclair's Nigel Searle said that the Boston show will closely follow the Federal Communications Commission's approval of the ZX81 (expected shortly).

Sinclair expects to have about 150 ZX81s available for hands-on demonstrations, where groups can try ZX81s and ask questions of Sinclair representatives. In addition, there will be a lecture/film presentation about computers and exhibits representing different ZX81 applications.

MICROCOMMANDER INTERFACE

Interface Technology of Des Plaines, IL, has resumed plans to adapt the TRS-80 Microcommander to the ZX80 (Dec.80,p.2). The device gives the computer direct control of up to 256 lights and appliances. Interface Technology's Edward March did not specify production dates.

RENEWAL BONUS

Get a free binder for your SYNTAX issues when you renew your subscription. Sturdy binder holds at least one year's copies. About 12 weeks before your subscription expires, you'll receive a white early renewal card in the mail. Return the card with your order within two weeks and we'll send your binder absolutely free. Remember, your order must be postmarked no later than 2 weeks after our postmark. Offer does not apply to later notices. Subscribers outside North America: we'll allow you an extra two weeks to reply. Remember, renew early!

ROM LIST DELAY

Our 4K ROM assembly listing has been delayed for a couple of weeks. We will accept orders at the prepublication price of $29 right up until the manuscript goes to our printer, so send check or credit card no. to SYNTAX, RD 2 Box 457, Bolton Rd., Harvard, MA 01451.

CONTINUOUS DISPLAY UTILITY

Want a flicker-free screen? JRS Software now offers software for continuous display. For more information, contact JRS Software, 19 Wayside Avenue, Worthing, England BN13 3JU.

ZX81 POCKETBOOK

The ZX81 POCKETBOOK is now available from Phipps Associates, 3 Downs Ave., Epsom, Surrey, England KT18 5HQ. We'll review it soon.

SYNTAX ERROR: *Rex Pedigo called to correct ZX80 Checkbook (July 81) instructions. Type GO TO 100, not GO TO 10, to run again.
*Dann Veldkamp of Ames, IA, noted that the address we calculated in BEGINNERS' LOADING ML PROGRAMS (Aug.81,p.11) was 17397, not 17399. The program works anyway, since the computer ignores the extra 2 bytes.
*Gene Richardson found an error in Al Salt's BUILD ADDITIONAL RAM (Mar 81,p.3). The 2nd paragraph, 3rd line, says "A11, A12, and A13"; it should be "A10, A11, and A12", as on schematic Figure 3.

Will Hiatt submitted the following changes to last month's IMPROVING DISPLAY program for 16K RAM:

```
 140 LET Q=PEEK(16396)+PEEK(1639
7)*246+375
 365 IF D=45 THEN LET P=Q
```

Line 140 adds flexibility to the display. Line 365 adds direction mode command H. H lets the cursor hold position, so you don't have to move beyond the display.

If you have a logic probe, you can record the high (H), low (L), and pulse (P) state on each pin of each chip when the computer is idling. When it malfunctions, check them again. You may be able to quickly find the problem area. Here is the idle state for my ZX80 IC 16 (where L or H is replaced by an asterisk, the probe lamp was glowing dimly).

Marty Irons, Goshen, NY

1	L	P	H		14			H
2	L				13	L	P	
3	L	P	H		12	L	P	H
4	L	P			11	L	P	*
5	L	P			10	L	P	
6		P	H		9	L		
7	L				8			H

(From QZX, newsletter for amateur ham operators, free. For info, contact Marty, K2MI, at 46 Magic Circle Drive, Goshen NY, 10924)

SINE FUNCTIONS ON 4K

You can closely approximate sine functions on your 4K ROM with a few BASIC statements.

These equations simulate sine between 0 and 45° (X in integer degrees):

$$1000\sin(X)=\left[\frac{2621X}{15}-\frac{11X^2}{450}-\frac{11X^3}{1350}\right]/10$$

Program sine using:

```
R=(11*X**2)/450+((11*X**2)/1350)*X
S=(262*X)/15-R/10
```

where S is 1000 times sin(X).

Lori Olson, Minneapolis, MN

OUR POLICY ON CONTRIBUTED MATERIAL

SYNTAX ZX80 invites you to express opinions related to the ZX80 and the newsletter. We will print, as space allows, letters discussing items of general interest. Of course, we reserve the right to edit letters to a suitable length and to refuse publication of any material.

We welcome program listings for all levels of expertise. Programs can be for any fun or useful purpose. We will test run each one before publishing it, but we will not debug programs; please send only workable listings.

In return for your listing, we will pay you a token fee of $2.00 per program we use. This payment gives us the nonexclusive right to use that program in any form, world-wide. This means you can still use it, sell it, or give it away, and so can we.

We will consider submissions of news and hardware or software reviews. Please keep articles short (350-400 words). Again, we reserve the right to edit accepted articles to a suitable length. We will pay 7 cents per 6 characters, including spaces and punctuation, for accepted articles.

When you send in programs for possible publication in SYNTAX, please include the following information:

- How to operate the program, including what to input if it does not contain prompts.
- Whether you can run the program over again and how.
- How to exit the program.
- The Syntactic Sum (using the Syntactic Sum program in the February, 1981, issue).
- Whether it fits in 1K or 2K RAM (or 16K when available).
- Whether it uses the 4K or 8K RAM.

We pay for this explanatory text at the same rate as for articles in addition to payment for the program itself.

If you want us to return your original program listing or article, please include a self-addressed, stamped envelope. Otherwise, we cannot return submitted material.

SYNTAX ZX80 is published monthly by Syntax ZX80, Inc., a wholly-owned subsidiary of The Harvard Group, RD 2, Box 457, Bolton Road, Harvard, MA 01451. Telephone 617/456-3661. Subscriptions: 12 issues, $25. Single issue, $4.

Publisher: Kirtland H. Olson
Editor: Ann L. Zevnik
Associate Editor: Susan G. Barber
Technical consultant: Eric K. Olson

Printed by Joseph E. Marhefka, Jr.
Clinton Offset Printers
Clinton, MA 01510

 ISSN 0273-2696

LANDING A LEM ON THE MOON

This ZX80 version sets a LEM 2,530 meters above the moon with only 300 liters of fuel. You must land without crashing or running out of fuel.

To play, type RUN (NL) and enter how many liters of fuel (0-75, inclusive) you wish to use. If you make a mistake on input, just reRUN. You've crashed if you land at more than 20 km/h.

Andre Bonja, Blainville, Quebec

```
  30 CLS
  40 LET M=75
  50 LET T=0
  60 LET X=2350
  70 LET V=0
  80 LET L=300
 130 PRINT "TIME :";T;" SECS"
 140 PRINT "ALTITUDE :";X;" METE
RS"
 150 PRINT "SPEED :";V;" M/S"
 152 IF V<0 THEN PRINT "DOWN :";
 154 IF V>0 THEN PRINT "UP :";
 155 IF V=0 THEN PRINT "STEADY:";
 156 PRINT (36*ABS(V)/10);" K/H"
 160 PRINT "FUEL :";L;" LITERS"
 170 IF L<0 OR L=0 THEN GO TO 20
00
 190 PRINT "FUEL ? ";
 200 INPUT L1
 205 PRINT L1
 210 IF L1>M THEN GO TO 2030
 230 LET L=L-L1
 300 LET T=T+1
 310 LET X=X+V-((5-L1)/2)
 315 IF V<-6 AND X<0 OR X=0 THEN
GO TO 1000
 317 IF X<0 OR X=0 AND NOT L<0 T
HEN GO TO 1005
 320 LET V=V-(5-L1)
 350 CLS
 360 GO TO 130
1000 PRINT "CRRRASH"
1002 GO TO 1020
1005 PRINT "TOUCH DOWN"
1010 PRINT "AT SPEED OF ";(36*V)
/10;" K/H"
1020 PRINT "STILL HAVE ";L;" LIT
ERS.";
1030 PRINT "  WANT TRY AGAIN? Y/
N"
1035 INPUT Z$
1040 IF Z$="Y" THEN GO TO 30
1999 STOP
2000 PRINT "ALARM TANK EMPTY."
2010 LET L1=0
2020 GO TO 1030
2040 PRINT "MAX= ";M;" L";
2050 GO TO 230
Syntactic Sum: -12334, 4K
```

THE MATH PROGRAM

The Math Program of Champaign, IL, presents tailored study, using ZX80s, to those who want extra help in math. Students range from kids needing help with fractions, to students wanting to move ahead, to adults with math anxiety.

The following ZX80 program by 11-year-old Eric Deidman multiplies a 1-by-4 number matrix by a 4-by-1 matrix to get a 1-by-1 matrix.

For more information on the Math Program, write Don Cohen or Jerry Glynn at 809 Stratford Drive, Champaign, IL 61820, or call 217/356-4761 or 328-1640.

```
   5 PRINT "INPUT NUMBERS TO BE
USED"
  10 INPUT A
  20 INPUT B
  30 INPUT C
  40 INPUT D
  50 INPUT E
  60 INPUT F
  70 INPUT G
  80 INPUT H
  90 LET J= A*E+B*F+C*G+D*H
 100 PRINT "THE ANSWER IS ";J
Syntactic Sum: 8791, 4K
```

DATA DUBBER

The Peripheral People now offer a device to help you LOAD and SAVE cleanly. According to PP's Tim Stoner, the Data Dubber draws its power from the ZX80 and includes outlets for connecting two tape recorders. For more information, contact The Peripheral People, PO Box 21123, Seattle, WA 98111.

ZX80 BUDGET

Now you can keep track of expenses using only 4K ROM, 1K RAM. As written, this program keeps addition totals on 18 accounts.

Type the program and save on tape. The first time you use it, type RUN (NL). When the prompt appears, enter 1 (NL), and set up your 3-digit account numbers (a 3-digit code for each account). Next, enter costs with account numbers and save everything on another tape. Remember to enter dollars and cents separately.

For later entries, LOAD, type GO TO 150 (NL), enter account numbers and costs, and save on another tape for next time. To see account totals, hit GO TO 290 (NL).

Quentin I. Smith, Kihie, HI

```
  10 DIM X(18)
  20 DIM Y(18)
  30 DIM Z(18)
  40 FOR J=1 TO 18
  50 PRINT "ENTER CODE"
  60 INPUT X(J)
  70 CLS
  80 NEXT J
  90 FOR J=1 TO 18
 100 PRINT X(J),
 110 NEXT J
 120 CLS
 130 PRINT "ENTER 1(NL) TO CONTI
NUE BUDGET ENTRIES"
 140 INPUT T
 145 IF T=1 THEN GO TO 150
 147 GO TO 290
 150 CLS
 160 PRINT " ENTER COST CODE"
 170 INPUT A
 180 PRINT "ENTER COST DOLLARS"
 190 INPUT B
 200 PRINT " ENTER COST CENTS"
 210 INPUT C
 220 FOR J=1 TO 18
 230 IF X(J)=A THEN LET Y(J)=Y(J
)+B
 240 IF X(J)=A THEN LET Z(J)=Z(J
)+C
 250 IF Z(J)>99 THEN GO SUB 1000
 260 IF X(J)=A THEN GO TO 120
 270 NEXT J
 280 IF J=19 THEN GO TO 2000
 290 CLS
 300 LET D=0
 310 LET E=0
 320 PRINT "CODE","AMOUNT"
 330 FOR J=1 TO 18
 340 PRINT X(J),Y(J);".";Z(J)
 350 LET D=D+Y(J)
 360 LET E=E+Z(J)
 370 NEXT J
 380 IF E>99 THEN GO SUB 4000
 390 PRINT "TOTAL",D;".";E
 400 STOP
1000 LET Z(J)=Z(J)-100
1010 LET Y(J)=Y(J)+1
1020 RETURN
2000 CLS
2010 PRINT "CODE NOT FOUND - RE
-ENTER"
2015 CLS
2020 GO TO 150
4000 LET E=E-100
4010 LET D=D+1
4020 IF E>99 THEN GO TO 4000
4030 RETURN
Syntactic Sum: -4597, 4K
```

FROM OTHER BASICS--VAL AND LEN

Microsoft BASIC is used by the Apple, TRS-80, and PET computers. Lots of software is available for these computers, so you can convert existing programs rather than write your own. This month we'll look at 2 Microsoft BASIC keywords not offered in 4K ZX80 BASIC--LEN and VAL. When you find these commands in Microsoft programs, just substitute ZX80 BASIC commands.

LEN(N$) finds the number of characters in string N$. To simulate LEN in 4K BASIC, we use TL$, which strips a string of its first character. For example,

```
 20 INPUT S$
 30 LET L=0
 40 LET D$=S$
 50 IF D$="" THEN GO TO 90
 60 LET L=L+1
 70 LET D$=TL$(D$)
 80 GO TO 50
 90 PRINT "THE STRING ";S$; HAS
A LENGTH OF ";L;"."
```

Line 20 asks for the string and stores it as S$. The computer uses L to count characters; line 30 tells the ZX80 to start L out at 0. D$ is a working copy of S$, to be stripped by TL$. Each time the computer uses TL$, it checks to see if there's anything left to the string, then adds 1 to L. This continues until D$ is empty. The above program in Microsoft BASIC looks like this (Don't enter this in your 4K ZX80):

```
 20 INPUT S$
 30 L=LEN(S$)
 40 PRINT "THE STRING ";S$;" HAS A
LENGTH OF ";L;"."
 50 END
```

VAL(S$) changes string numerals to numbers. The ZX80 distinguishes NUMBERS, mathematical quantities, from NUMERALS, written representations of mathematical quantities. That is, "4" (Arabic) and "IV" (Roman) are both numerals meaning the square of 2 (a number). The computer stores numerals in quotation marks differently than it does numbers, so it can't manipulate string numerals the same as it can numbers. Try the following VAL routine in 4K BASIC:

```
 10 INPUT S$
 20 LET V=0
 30 LET L=0
 40 LET D$=S$
 50 IF D$="" THEN GO TO 90
 60 LET L=L+1
 70 LET D$=TL$(D$)
 80 GO TO 50
 90 LET D$=S$
100 FOR I=1 TO L
110 LET V=V+(CODE(D$)-28)*10**(
L-I)
120 IF I<L THEN LET D$=TL$(D$)
130 NEXT I
150 PRINT S$;" EQUALS ";V;"."
```

The first half of this routine is identical to our LEN program. Then, we use character codes to determine the number represented by each digit. Next we multiply it by a power of 10 determined by each digit's position in the string.

The computer cannot, for instance, multiply a string by a number. VAL allows you to use numerals in strings, then convert them into numbers for the ZX80. The equivalent routine in Microsoft would be (again, your 4K machine won't like this!):

```
 10 INPUT S$
 20 LET V=VAL(S$)
 40 PRINT S$;" EQUALS ";V;"."
 50 END
```

Richard Forsen, New Hartford, NY

Would you like a ZX80 interface to the outside world? Joseph Hartmann of Jersey City, NJ, is considering developing an STD BUS card for the ZX80. For more information, write him at 101 Tonnele Ave. Jersey City, NJ 07306.

DEAR EDITOR:

I've noticed an interesting feature on my 4K ROM: when I enter PRINT USR(623) (NL), I get a clear screen with a solid cursor in the lower left-hand corner. Then if I try to enter lines of program text, I get a "bar code" effect displayed instead of alphanumeric symbols.

The Z80 machine language routine that I actually run loads the I (interrupt) vector register with the value in the accumulator. Using my own value instead of that in the accumulator, I obtain other strange values and characters on the screen. These routines have no effect on actual programming, but seem to change only the video pattern on the TV screen.

David Shulman, Peabody, MA

For those who want to program with this bar code generator, only NEW (NL) wipes it out. When I printed numerals 0 through 9, I got logical bar sequences that could probably be easily read by the ZX80 or an outside device.--SB

With a tape recorder and 1K RAM, I have flawless recording and play-back with both ear and mic cables attached at all times. With the 16K RAM attached, however, a hum develops about halfway into the 5-second silence, ruining the tape for further LOADing.

Robert Keller, Rochester, MN

We've had the same trouble, but not steadily. First of all, make sure you've hooked up the right power supply. Next, use a pencil eraser to clean the connector's finger contacts--grime can cause humming. On our 8K machines, the 16K RAM appears to overburden its own power supply as well as the standard one, so we hooked up power through a transformer and adjust voltage back to 9V whenever it dips.--SB

Is there a way of modifying the ZX80 so that the cursor could traverse a line when holding the key down? Is there a loading effect problem if one wants to bypass the RF modulator (Ch. 34) for direct video for a monitor?

R. Allan Smith, Sunland, CA

For direct video, try a high-impedance monitor. A 50-75 Ω monitor needs an emitter-follower as described in Sinclair's technical manual. As for cursor movement, Softsync's games use continuous cursor movement, but we haven't read the code yet.--SB

Responding to an INPUT statement with EDIT isn't fatal! About all that happens is that the screen has been cleared and the current line written into the edit buffer. Just use the cursor and RUBOUT keys to wipe out the copy of the current line and type in valid input. If the INPUT statement was for a string, you must, of course, put in the quotes--but at least you don't have to pull the plug.

Jim Williams, Calumet City, IL

I've had interference problems with the TV display since purchase of my ZX80. The ZX80 is connected to a 12-inch black-and-white TV, but I've had no success with the fine-tune adjustment. I also tried removing nearby metal objects.

Richard Nunley, Moline, IL

Try playing with the horizontal hold. The contrast-brightness combination also makes a big difference, and this varies from set to set. Your problem may be RF interference--see Herb Sturges' hint on page 9. Herb also noticed poor soldering in his video cable had melted plastic insulation separating the signal carrier and ground. Take a look.--SB

The following SYNTAX readers wish to contact others in their areas. To reach users in your area, send us your name and address.

*Jay McFarling, 1327 Vista Verde Way, Escondido, CA 92027

*John R. Mullen, 8578 Terrang Ct., Rockford, IL 61111

*Bill Paige, United Press International, 360 North Michigan Ave., Chicago, IL. 312/781-1640

*Bob Routson, 2114 Randolph Rd., Silver Spring, MD 20909

*Mel Routt, PO Box 596, Safety Harbor, FL 33572

*Steve Singer, 14032 Halstead Ct.,Tampa, FL 33612 813/977-1150

POPULAR ELECTRONICS plans to start a special interest group service on Compu-Serv and The Source. If you belong to a ZX80 or other computer club and would like your club calendar published, or wish to start a new ZX80 group, write POPULAR ELECTRONICS, 1 Park Ave., New York, NY 10016, 212/725-3568.

MANIPULATING STRINGS

These routines broaden your ability to use strings in interesting ways. STRING REVERSAL (#1) prints strings backward in inverse video characters and is limited to the number of elements in array C (here, 31 characters). To increase the number of elements, just increase C up to memory limits. INVERSE GRAPHICS (#2) converts strings up to 10 characters long into inverse video.

```
   5 DIM C(30)
  10 PRINT "ENTER A STRING"
  20 INPUT A$
  30 PRINT
  40 PRINT "REVERSE OF ";A$;"  IS
:"
  50 LET I=0
  60 LET I=I+1
  70 LET C(I)=CODE(A$)
  80 LET A$=TL$(A$)
  90 IF NOT A$="" THEN GO TO 60
 100 FOR N=0 TO I-1
 110 LET K=I-N
 120 PRINT CHR$(C(K));
 130 NEXT N
 140 PRINT
 150 PRINT
 160 FOR N=0 TO I-1
 170 LET K=I-N
 180 PRINT CHR$(C(K)+128);
 190 NEXT N
 200 PRINT
 210 PRINT
 220 PRINT "-END-"
Syntactic Sum:  23666, 4K
```

Austin Brown, Jr., Golden, CO

NOTE: Do not use "+" in A$.

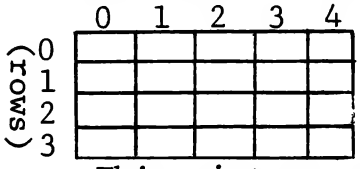
```
  10 LET A$="INVERSE GRAPHICS"
 999 REM **INVERSE SUBROUTINE**
1000 LET ZZ=0
1001 LET XX=ZZ+16431
1002 IF PEEK(XX)=1 THEN LIST
1003 POKE XX,128+PEEK(XX)
1004 LET ZZ=ZZ+1
1005 GO TO 1001
Syntactic Sum:  12223, 4K
```

Peter Herschel, Newtown Square, PA

PRINTING ON THE 8K ROM

One of the 8K ROM's nicest features is its increased printing capability. We have 2 new PRINT-like commands, PRINT AT and PLOT (with its corresponding UNPLOT).

Before we can understand how to use these commands, we should look at the ZX80 display screen. Regardless of what size TV you use, your 8K machine displays 32 columns across and 22 rows down. If you think of the screen as a grid, with lines between each column and each row, the grid forms little boxes, or screen elements. You can name each box uniquely, using its column and row number, like a graph:



This picture shows the 20 boxes at the top left corner of the screen. The origin, or beginning of the grid, is at the top left corner. To use PRINT AT, you must know the column and row numbers of where you want to print. For example, to print a single character at the dead center of any line, figure out the middle column number and choose the line you want to use. Each row has 32 columns, so the exact center of the line is at 15 1/2. We can only PRINT AT complete boxes, so we must choose either 15 or 16. If we want to print the letter A in the middle of line 6, we would type:

```
PRINT AT 6,16;"A"
```

(AT is F C; hit SHIFT NL, then C.) This means print 16 positions from the left and 7 lines from the top. The first line is numbered 0, as is the first column, so line 6 is the seventh down. (Use the form PRINT AT line,column.)

Try this sample program to see how PRINT AT can move the locations of characters:

```
 10 PRINT "ENTER THE COORDINATE
S FOR THE SAMPLE PHRASE.  X=";
 20 INPUT X
 30 PRINT X
 40 PRINT "Y=";
 50 INPUT Y
 60 PRINT Y
 70 PRINT "ENTER SAMPLE PHRASE"
 80 INPUT A$
 90 CLS
100 PRINT AT X,Y;A$
RUN
```

This program asks for the coordinates you want and the words you want to print (lines 10-70). Line 80 allows you to choose the sample phrase. Line 90 clears that display from the screen. Line 100 tells the computer to print your string (the phrase you typed) at the location named by the coordinates X and Y. You can use string variables (like A$ in our example) or literal variables in PRINT AT statements. (Literal strings are those you enclose in quotation marks in PRINT statements, like PRINT "ABC".) You can type the coordinates for X and Y as numbers (4,23 for example) or as variables (as in our sample program).

PRINT AT commands allow you to put words or characters anywhere on the display without using up lots of RAM space. To put CAT on the fifth line with blank lines above it, for example, requires five PRINT statements with the 4K ROM (4 blank PRINT lines and one PRINT "CAT"). The 8K ROM needs only one PRINT command to do this.

But PRINT AT has limitations. You cannot PRINT AT beyond the edges of the screen, so don't ask the computer to print a character that would run over the edge. Run our sample program, changing X and Y to see which boxes are useable. You also can't print on lines 22 and 23 (bottom two).

PLOT and UNPLOT also use individual screen elements, but smaller ones. Suppose we cut each box into 4 quarters. Each quarter is called a pixel (short for picture element). See the ZX81/8K manual, p.123, for a picture of the

pixel grid. Note that the grid begins at the lower left of the screen, not the upper left.

Cutting up the boxes gives us twice as many possible rows and columns. Instead of going from 0 to 31, the columns go from 0 to 63 (giving 64 total columns). The rows now go from 0 to 43 (giving 44 total rows). So (63,43) is at the top right of the screen; (0,0) is at the bottom left.

To use a PLOT or UNPLOT command, you again specify the X- and Y-coordinates of the location. PLOT and UNPLOT use pixels instead of whole character spaces, so our X- and Y-coordinates name precisely which quarter of a character space we wish to use.

PLOT X,Y blackens the pixel at the column X and row Y. UNPLOT X,Y unblackens that same pixel, as you would expect. Use the form PLOT column,line.

You can use numbers for the coordinates, like 4,65. Or you can use numerical expressions such as 3*SQR 79 (3 times the square root of 79). You can also use numerical variables, like A or B, as we did in our PRINT AT program above.

PLOT and UNPLOT are useful for graphs of numerical functions. See the 8K ROM manual, pp.119-122, for examples of graphs.

Herb Sturges called us with a way to cut RF interference to the ZX80. He found that the rivets holding the upper and lower halves of the computer casing didn't exert enough pressure to keep them together. RF radiation leaks through rivet holes and between ridges in the rear near ZX80 connectors.

Herb repainted ZX80 mating attachment flanges with printed circuit repair paint to reseal the electrical shielding. He explained that the metallic film improves contact sufficiently to prevent the unusual changes in the display occurring when you apply pressure to the case.

RANDOM FUNCTIONS IN 4K, 8K

Several readers have asked us for clarification of ZX80 random functions. This month we'll compare RND (4K, 8K), RAND (8K), and RANDOMISE (4K).

All three functions in both ROMs make pseudo-random numbers. The ZX80 generates a long number sequence in memory, so long that the sequence appears random. Some functions allow you to tell the computer where in the sequence to start picking numbers. Otherwise, system variables determine start.

RND is probably the most confusing when translating from 4K to 8K ROM. In 4K, RND takes the form RND(X), where X is an integer. RND(X) yields a "random" integer (whole number) between 1 and X. In 8K, RND uses no X and picks decimal fractions between 0 and 1.

RAND Y sets the randomness of 8K RND. (That is, RAND Y will pick numbers with Y between 1 and 65535.) For instance, RAND 25 will give you the 25th number in the ZX80's sequence. Each Y will give you the same "random" number between 0 and 1 in 8K every time.

RANDOMISE has two different functions in 4K. RANDOMISE N points to the number sequence the same way as RAND Y does for 8K. However, RANDOMISE (no N, or N=0) picks numbers depending on how long the machine has been on.

```
RANDOMISE N = RAND N, N <>0
RANDOMISE.RANDOMISE 0 = RAND 0
```

Lori Olson added this note to her sine and cosine programs. To approximate tangent with 4K ROM, use this equation:

$$1000\tan(X)=1000\frac{\sin(X)}{\cos(X)}$$

Program tangent using:

```
LET T=(25*S)/(C/40)
```

where S=1000sin(X), C=1000cos(X). (See p.2 & p.16 for sin and cos.) The error increases with X and any X>88 produces an overflow.

READER'S REPORT ON MICROACE VIDEO

Not for neophytes--that's John Strain's assessment of the MicroAce Flicker-Free Video Add-on Board. According to John, "The flicker-free mod circuit board does not physically fit inside the ZX80 case. The components supplied completely prevented mounting the circuit board inside the ZX80." Also, you can't use the top display line; you must add components to make the circuit work properly and modify a piece of the ZX80 board not shown on Sinclair's schematic.

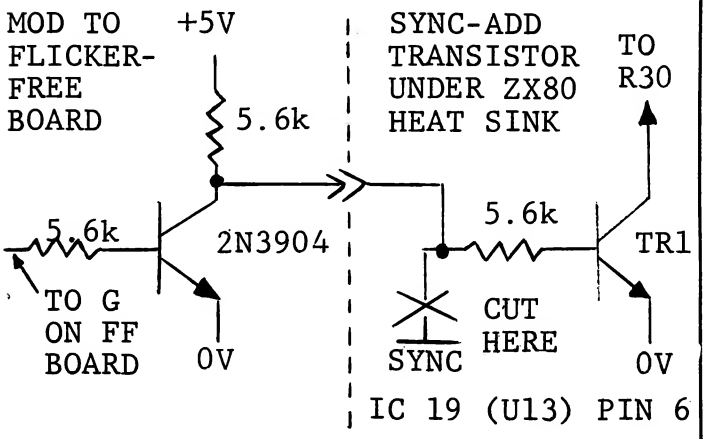
To fit the MicroAce mod in the Sinclair case, John reports: "After acquiring miniature components and cutting and sanding leads on both the flicker-free kit and the area of the ZX80 where I was going to mount the mod, I was finally able to get it inside the ZX80 case, under the ZX80 circuit board. Even then, the bottom of the ZX80 case bulged out, causing it to rock as I pressed the keys."

As supplied, the modification forced the TV to re-synchronize at every frame. As MicroAce says in their instructions, "There are a couple of funnys with this addon in particular the top line of the display slants to the right a bit and R3 (1K) sometimes needs to be 470 Ohms to centralize the characters in there field! (sic)" John found this unacceptable and worked out the scheme that follows.

You must add a simple, one-transistor circuit to the mod kit. This provides the $\overline{\text{SYNC}}$ signal needed by the ZX80 (but not shown on the Sinclair schematic). You add only three parts, but the PC

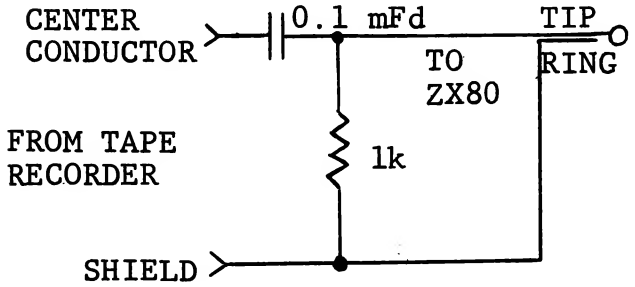
board provides no room for them.

Your ZX80 needs one more trace cut and one more connection to the modified flicker-free kit to accept this change. John says: "The sync-add transistor is located to the left of the modulator unit and under the voltage regulator heat sink. Cut the printed circuit going to the 5.6k Ohm resistor which feeds the base of the sync add transistor. Wire the collector of the flicker-free mod's added transistor to the 5.6k Ohm resistor part of the cut circuit."



HARDWARE REPORT

Brian O'Brien of Weston, MA, called with a solution to problems with LOAD. None of 3 recommended brands of tape recorder worked well on his ZX80, so he put a 0.1 uF capacitor in series with the input on the ear line and a 1k resistor across the input (this might change with the tape recorder).



All 3 recorders now work. Brian added that most tape recorders put out 20-200 Hz noise, and he didn't think the ZX80's low-frequency filter was adequate. He also suggests that readers try using a 5k variable resistor and changing values until they get one that works well.

Brian also suggests using an external regulated DC power supply to run the recorder. Most small recorders that operate on batteries will run from a 5-volt regulated supply. Because the recorders we checked drew less than 400 mA, a 5V, 0.5 A supply would suffice. This eliminates an alternate path for motor noise to reach the ZX80.

PRIME NUMBERS

Numbers that cannot be divided evenly by numbers other than 1 or themselves are termed prime. This program lists all prime numbers between 1 and 1000.

Type RUN (NL), and the screen will go blank as the ZX80 figures out the numbers. The program takes a while to finish; press BREAK to list what numbers the computer has found so far.

Phyllis Richmond, Cleveland, OH

```
  10 PRINT "PRIME NUMBERS"
  20 FOR N=1 TO 999
  30 FOR K=3 TO 31
  40 LET Q=N/K
  50 LET R=N-Q*K
  60 IF Q=1 THEN GO TO 200
  70 IF R=0 THEN GO TO 250
  80 IF N=1 AND K=31 THEN GO TO
400
  90 LET K=K+1
  95 NEXT K
 200 PRINT " ";N;
 250 LET N=N+1
 300 NEXT N
 350 STOP
 400 PRINT "1 2";
 410 LET N=N+1
 420 NEXT N
Syntactic Sum: 16764, 4K
```

READING MACHINE CODE ASSEMBLY

Beginning machine language (ML) programmers sometimes have trouble deciphering all the numbers that make up an assembly listing. Let's take a look at a few lines (we've ruled off the columns).

00F7	FEC0	206	LL7	CP	0C0H
00F9	EA0801	207		JP	PE, LL8

First of all, let's mark off the characters in column 2 by pairs (FE C0 EA 08 01). Each pair is a hex byte and resides in a separate memory address.

Column 1 lists the memory location (or address) at which the first ML byte on that line resides. For example, address 00F7h (h stands for hex) contains ML code FEh (column 2).

Where are the other bytes? Since each byte takes up 1 address, add 1 to column 1 for every digit pair you count over to the right. For example, byte #5, 01h (column 2, line 2) is 2 bytes to the right of byte EAh. Remember to add in hex. Byte 01h resides in address 00FBh (00F9h+2h).

When you type lines into an ML assembler, you must enter numbered program lines so the computer knows which line comes before what. Your line numbers take up column 3.

Optional subroutine labels go in column 4, and ML mnemonics (program lines) appear in column 5. Here, the first line in subroutine LL7 reads "CP 0C0H". The computer doesn't execute these lines; they help the programmer know what the code in column 2 means.

To help others decipher their work, programmers add a 6th column (not shown here) for comments. You can distinquish ML comments (like REM in BASIC) because they're preceded by semicolons. Sometimes comments will start to the left of column 6, but computers consider any text to the right of a semicolon to be a comment.

MACHINE CODE PRINT ROUTINE

Machine language (ML) permits you to print characters anywhere on the screen. Harness the ZX80's own print routine with RST 0010.

RST (ReSTart) 0010 is essentially the same in both 8K and 4K ROMs. It saves contents of the PC register (program line counter) in the stack, loads 0010h into PC, and branches to address 0010h.

Initially, the A register must hold the code for the symbol to be printed, and you must supply row and column numbers for the print position. To use in 4K, load position coordinates into the B' and C' registers. For 8K, load values into system variable S-POSN (column value in address 16441, row value in 16442). Variable DF-CC must contain the address in the display file for these coordinates.

Try running this 4K print program using RST 0010:

```
    10 REM YSNOT YYNOT YNNOT YTNOT
YANOT YXNOT Y NOT TAN
    20 LET L=USR 16514
    30 GO TO 20
```

(NOTs and TAN are function keys)
Line 10 translates into:

Hex	Mnemonic
3E53	LD A,'S'
D7	RST 0010h
3E59	LD A,'Y'
D7	RST 0010h
3E4E	LD A,'N'
D7	RST 0010h
3E54	LD A,'T'
D7	RST 0010h
3E41	LD A,'A'
D7	RST 0010h
3E58	LD A,'X'
D7	RST 0010h
3E20	LD A,' '
D7	RST 0010h
C9	RETURN

(Beginners--this assembly listing contains only cols. 2 & 5.)

Ian Logan, Skellingthorpe, England
(The 4K routine called by RST 0010 begins at 0560h. We include it here for convenience.--SB)

```
RST 0010 JUMP TO SENDCH
0076            00110 NL      EQU     76H
0020            00120 L8LEN   EQU     20H
0958            00130 CHK8L   EQU     0958H
                00140 ;WRITE CHARACTER FROM A TO DISPLAY FILE
                00150 ;REQS B' = LINE #  (TOP LINE IS 24,BOTTOM IS 1)
                00160 ;C' = 1 + # POSNS LEFT ON LINE (VIZ 33 AT LH END, 2 AT
                00170 ;RH, IF LINE FULL, ZERO IF MUST GO ON NEXT LINE)
                00180 ;D'E' -> ADDRESS TO WHICH TO WRITE:UPDATES B'C', IF
                00190 ;NO ROOM IN RAM OR ON SCREEN RETURNS NC, ELSE WRITES
                00200 ;CHAR & MOVES D'E' ON & SETS CARRY: USES H'L',
                00210 ;PRESERVES BCDEHL, SETS PO: "INTERFACE AS SENDCH"
                00220 ;MEANS B'C'D'E'H'L' AND CARRY ARE USED AS ABOVE--
                00230 ;DOES NOT IMPLY ANYTHING ABOUT ABCDEHL OR P/V
0560 D9         00250 SENDCH  EXX
0561 67         00260         LD      H,A
0562 17         00270         RLA
0563 17         00280         RLA             ;GET D6 OF A INTO CARRY FLAG
0564 0D         00290         DEC     C
0565 3002       00300         JR      NC,LSC1
0567 0E00       00310         LD      C,0
0569 FA7405     00320 LSC1    JP      M,LSC2  ;IF 1ST CHAR ON LINE
056C 380E       00330         JR      C,LSC5  ;IF CHAR IS NEWLINE
056E 200C       00340         JR      NZ,LSC5 ;UNLESS 41ST CHARACTER
0570 3E76       00350         LD      A,NL
0572 12         00360         LD      (DE),A
0573 13         00370         INC     DE
0574 3802       00380 LSC2    JR      C,LSC3
0576 0E20       00390         LD      C,L8LEN
0578 A7         00400 LSC3    AND     A
0579 05         00410         DEC     B
057A 2806       00420         JR      Z,LSC6
057C 68         00430 LSC5    LD      L,B
057D CD5809     00440         CALL    CHK8L
0580 12         00450         LD      (DE),A  ;WRITE CHAR
0581 13         00460         INC     DE
0582 D9         00470 LSC6    EXX     ;SETS PO (FOR NV) IN EITHER CASE
0583 C9         00480         RET
```

KNOW YOUR GAS HOGS

Use your ZX80 to compare fuel economies and estimate total gas expenditures. This program for 8K ROM, 1K RAM calculates yearly fuel costs for any two vehicles. You enter miles driven (actual or projected) per year, fuel price, and gas mileage per gallon. It returns the total fuel cost per year for each car.

Mel Routt, Safety Harbor, FL

```
  10 PRINT "MILES = ";
  15 INPUT M
  20 PRINT M
  25 PRINT "PRICE/GAL = ";
  30 INPUT P
  35 PRINT P
  40 PRINT "VEH 1 MPG = ";
  45 INPUT V1MPG
  50 PRINT V1MPG
  55 PRINT "VEH 2 MPH = ";
  60 INPUT V2MPG
  65 PRINT V2MPG
  70 LET V1G=M/V1MPG
  75 LET X=V1G
  80 GOSUB 500
  85 LET V1G=X
  90 LET V2G=M/V2MPG
  95 LET X=V2G
 100 GOSUB 500
 105 LET V2G=X
 110 LET V1D=V1G*P
 115 LET X=V1D
 120 GOSUB 500
 125 LET V1D=X
 130 LET V2D=V2G*P
 135 LET X=V2D
 140 GOSUB 500
 150 PRINT
 155 PRINT "VEH:   GAL:   COST;"
 160 PRINT
 165 PRINT " 1     ";V1G;"  $ ";
V1D
 170 PRINT
 175 PRINT " 2     ";V2G;"  $ ";
V2D
 180 STOP
 500 LET X=(INT (100*(X+.005)))/
100
 505 RETURN
Syntactic Sum:  28144, 8K
```

BOOK REVIEW

Title: Problem-Solving Principles for Programmers: Applied Logic, Psychology and Grit
By: William E. Lewis
From: Hayden Book Co., Rochelle Park, NJ
Price: $9.95 (paper)

Lewis' book "consists of three interwoven conceptual threads: general problem solving, program problem solving, and the influence of psychology on the overall problem-solving process."

I liked the book and recommend it to people who seek either an introduction to, or increased breadth in, problem solving. You'll find the style readable and the INTERLINGUA pseudo-code helpful. Lewis uses both non-computer and computer examples; several examples use classic problems--penny-weighing, river-crossing, and black and white hats. In addition, he briefly discusses top-down design concepts for programming.

You are not alone, other people must also labor to solve problems on (or off) the computer. This book will teach you new tricks if you're inexperienced or remind you of old ones if you've been at this a while. When you get stuck, thumbing through this book could give you just the nudge you need.

INTERLINGUA, a computer-like language, reveals problem structures better than does BASIC. As you see from the example, you can translate to any computer language.

```
PROGRAM: TO COMPUTE GREATEST COMMON DENOMINATOR (GCD)

  input FIRST_NUM, SECOND_NUM
  while FIRST_NUM  not equal SECOND_NUM do

     if FIRST_NUM greater than SECOND NUM then
        subtract SECOND_NUM  from FIRST_NUM

     else

        subtract FIRST_NUM  from SECOND_NUM

  set GCD  to  FIRST_NUM
  print 'GREATEST COMMON DIVISOR:',GCD

end  * program *
```

In fact, Lewis offers one example programmed in 5 different languages, including IBM 5110 BASIC.

Since Sinclair BASIC differs from others in many details, I recommend the general book (Hayden 5138-7) for ZX80 owners. The BASIC version of the book uses READ and DATA extensively, and omits the valuable INTERLINGUA listings.--KO

ZX80 TABLE

This routine for 4K ROM, 16K RAM constructs number tables with columns lined up. You enter the largest number of digits that occur in your list, and the ZX80 figures out how many columns will fit across the display. Then, type in numbers one by one. The computer will line the numbers up by their rightmost digit.

R. Bharath, Marquette, MI

```
  30 PRINT "PROGRAM ARRANGES"
  40 PRINT "LIST OF INPUT NUMBER
S AS TABLE"
  50 PRINT "USER CHOOSES NUMBER
OF ROWS, COLS"
  60 PRINT
  70 PRINT "WHAT IS LARGEST NUMB
ER"
  80 PRINT "OF DIGITS IN ANY NUM
BER"
  90 PRINT "IN YOUR LIST?"
 100 INPUT L
 110 CLS
 120 LET F=L+1
 130 PRINT "IN THAT CASE THE LIM
IT IS"
 150 PRINT (32/F)-1,"COLUMNS"
 160 PRINT "HOW MANY ROWS?"
 170 INPUT R
 180 CLS
 190 PRINT "HOW MANY COLUMNS?"
 200 INPUT C
 210 CLS
 220 IF NOT C>(32/F) THEN GO TO
250
 230 PRINT "TOO MANY COLS. TRY A
GAIN."
 240 GO TO 70
 250 LET T=R*C
 260 DIM A(T)
 270 PRINT R,"ROWS",C,"COLS"
 280 PRINT "INPUT NUMBERS ONE BY
ONE"
 290 PRINT "ROW BY ROW"
 300 PRINT "REMEMBER ";L;" DIGIT
S OR LESS"
 310 FOR I=1 TO T
 320 INPUT A(I)
 330 NEXT I
 340 CLS
 350 FOR I=1 TO R
 360 FOR J=1 TO C
 370 LET K=C*(I-1)+J
 380 GO SUB 600
 390 PRINT A(K);
 400 NEXT J
 410 PRINT
 420 NEXT I
 430 PRINT
 440 PRINT
 450 PRINT "INPUT LIST ARRANGED
AS TABLE"
 460 PRINT R;" ROWS ";C;" COLS"
 470 STOP
 600 REM COUNT DIGITS FOR ALIGN
ING
 610 LET D=0
 620 LET G$=STR$(A(K))
 630 LET D=D+1
 640 LET G$=TL$(G$)
 650 IF NOT G$=CHR$(1) THEN GO T
O 630
 660 FOR S=D TO F
 670 PRINT " ";
 680 NEXT S
 690 RETURN
Syntactic Sum:  -6886, 4K
```

Extend the method of calculating sine and cosine on 4K (pp.2&16) to other slowly varying functions. Use cubic approximations. Example: for sine, 4 X-values equally spaced on the curve create 4 equations:

$10000\sin(0)=a+b(0)+c(0)^2+d(0)^3$
$10000\sin(15)=a+b(15)+c(15)^2+d(15)^3$
$10000\sin(30)=a+b(30)+c(30)^2+d(30)^3$
$10000\sin(45)=a+b(45)+c(45)^2+d(45)^3$

Solve these to get a general equation: $10000\sin(X)=a+bX+cX^2+dX^3$ Divide both sides by 10 for greater accuracy. To avoid overflows, multiply and divide carefully when programming the resulting equation.

CLASSIFIED ADS

Got something to sell or swap? Or are you looking for something special? Use SYNTAX classifieds. Reach hundreds of other readers for just $2.75 per line (4-line min.). Send your ad, typed 35 characters per line, with payment by 15th of any month for next month's issue to Classified Ads, SYNTAX, RD 2 Box 457 Bolton Rd., Harvard, MA 01451.

Homemade 46-pin female connectors for your ZX-80. $3 postpaid from Martin Irons, 46 Magic Circle Drive Goshen, NY 10924. Supply limited.

Nov/Dec SYNTAX combined reprints, $5, some other back issues available,$4 each. Send check or credit card no. to SYNTAX, RD 2 Box 457 Bolton Rd., Harvard, MA 01451.

Disassembler Program Cassette for 8K ROM/16K RAM. Display in 3 sec. Plus hex monitor, register & flag display. Read, write, debug Z80 code. Easy to use. Just $15 ppd. Sinware, Box 323, Dixon, NM 87527

Sinclair's complete 4K ROM listing available for $40. Send check or credit card no. to SYNTAX, RD 2 Box 457 Bolton Rd., Harvard, MA 01451

Upgrade your ZX-80 to 2K RAM. Not a kit, but an internally installed RAM expansion. Send your ZX-80 & a check for $27.95 to: Dave Frey, 619 Avignon Dr, Melbourne, FL 32935

2K/4K RAM:2Kx8 CMOS IC's on sockets replaceable w/2716,2732,2764 & 2532 EPROMs-mix or match RAM & EPROM. Inquire about PROM Programmer. Board-assembled w/2K-$30,w/4K-$45, board w/connector -$15. Add $2.50 for postage. Send check to: SABRE, 1415 Creek Hollow,Seabrook TX 77586

COSINE FUNCTIONS ON 4K ROM

Here's a way to approximate cosines with your 4K ROM (see p.2 for sine functions).

This equation simulates a cosine curve:

$$1000\cos(X)=[10000+32X-121X^2+34X^3]/10$$

Program cosine using:

```
B=-(((X*121)/15)*X)/5+((X**2/3)*X
  /45)*34/75
C=(10000+32*X/45+B)/10
```

where C is 1000 times cos(X).

For angles greater than 45°, use the relation cos(X)=sin(90-X). These equations and the ones on p.2 give sines and cosines within 3 in the least significant digit.

Lori Olson, Minneapolis, MN

SYNTAX

THE HARVARD GROUP
Bolton Road, Harvard, Mass. 01451

First Class